ZENITH OAR: Week One

The Tech Club - Robotics Team

Before moving on with the course, let us look at the goal/objective of the course.

Objective

Our objective in this course is to learn how we can design a robot that is capable of driving around obstacles to reach a previously known goal location. Now this might seem pretty straight-forward for humans. Robots need various information of the environment which are gathered using various sensors, in order to achieve its designated goals.
Now the approach we shall take in this course is called behavior-based robotics. In this approach, we shall try to identify multiple behaviors required to achieve our goal. For instance, one such behavior is avoiding obstacles in the environment while moving. Another example of a behavior is moving towards the target. When we separate the actions taken by the robot as behaviors, we can easily choose a behavior to be enacted given an environmental setting.

Now moving on with the course...

Differential Drive Robots

There are numerous models of mobile robots available. One common model is the differential drive robot. We will be working with a differential drive robot in our simulator. In order to be able to control the robot, we must understand how the model works and be able to model it as a system.

In this model, the robot consists of two drive wheels mounted on a common axis, and each wheel can independently be driven either forward or backward. When the wheels are driven in the same direction the bot moves forward or backward. However when the wheels are driven in opp directions, the robot

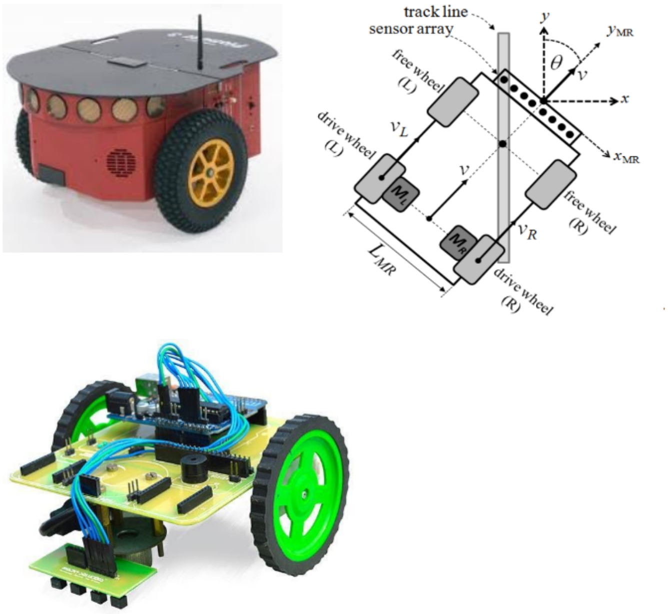

Figure 1: Differential drive robots

turns to the left or to the right. Turning can also be achieved by moving either of the wheels slower than the other.

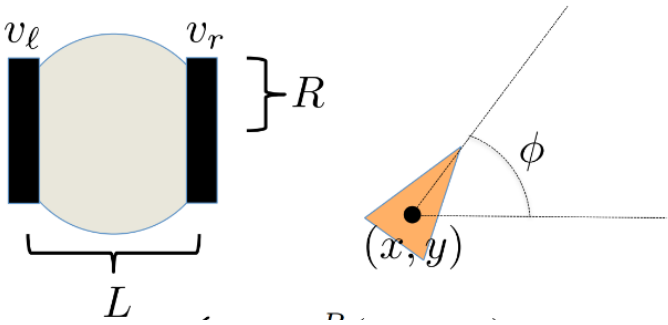

Figure 2: Differential drive robot and unicycle model

Let us consider the position of the robot using 3 coordinates, namely x,y and the heading angle (phi). The differential equations that describe the robot then can be given by the following equations.

$$\dot{x} = \frac{R}{2}(v_r + v_l)\cos\phi$$

$$\dot{y} = \frac{R}{2}(v_r + v_l)\sin\phi$$

$$\dot{\phi} = \frac{R}{L}(v_r - v_l)$$

Visit the following link to understand why these equations hold. [13.1.2.2 A differential drive (uiuc.edu)]

Now while designing control systems for such a model, it is not natural for us to think in terms of angular velocities of the wheels. Instead we plan paths in terms of an easier way to envision the model called the unicycle model.

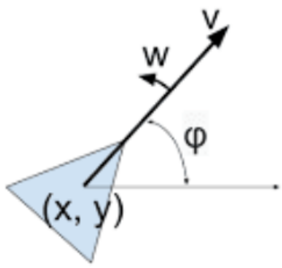

Figure 3: Unicycle model control

In a unicycle model, the control signals are linear velocity(v) and angular velocity(ω)$.Thevelocitiesareconsideredatthemidpointoftherobot'swheelaxis.$

Now we can use the unicycle model for our controllers, but our physical bot is a differential drive robot. So we need a way to relate the two models. The equations below show how the angular velocities of each wheel relate to the linear and angular velocities.

Inputs:

$$v$$

$$\omega$$

Dynamics:

$$\dot{x} = v \cos \phi$$

$$\dot{y} = v \sin \phi$$

$$\dot{\phi} = \omega$$

Comparing with differential drive model equations, we get -

$$v = \frac{R}{2}(v_r + v_l) \quad and \quad \omega = \frac{R}{L}(v_r - v_l)$$

Solving for left wheel and right wheel velocities ,we get -

$$v_r = \frac{2v + \omega L}{2R}$$
$$v_l = \frac{2v - \omega L}{2R}$$

Now with these equations we can easily send control commands to the robot and control it in the simulator. Now let us try just that...

Assignment

This week's assignment will consist of two tasks. You will have to try to control the pioneer-3 robot on Coppeliasim using the scene provided to you. You can make use of the Python API or MATLAB API. A sample code is provided on the Github repository to show an example of how to use the API.
The two tasks are mentioned below.

- To control the robot using only the differential drive model
- To implement both the models we saw above and transform from unicycle model to differential drive model

For the first task, you'll have to provide inputs in terms of the individual angular velocities of the wheels of the robot. For the second task, you'll have to provide inputs in terms of linear and angular velocity of a point on the robot. You can provide the input to your program however you wish, either via an interactive GUI (Graphical User Interface) or via the terminal.

Sensors

Let us talk about how a robot can sense its environment and its own actions. Just like how humans rely on their five senses (Sight, Sound, Smell, Taste, and Touch) to gather information about their surroundings, robots rely on its sensors to gather information about changes in its environment, whether or not it has performed an action correctly and so on. In our case, where our goal is to design a robot to avoid obstacles, we would need sensors to detect those obstacles. For this purpose, range finders and proximity sensors are used. A few sensors that can help robot to detect obstacles are IR sensors, Ultrasonic Sensors, LiDAR (Light Detection And Ranging), RGB-D cameras, to name a few.

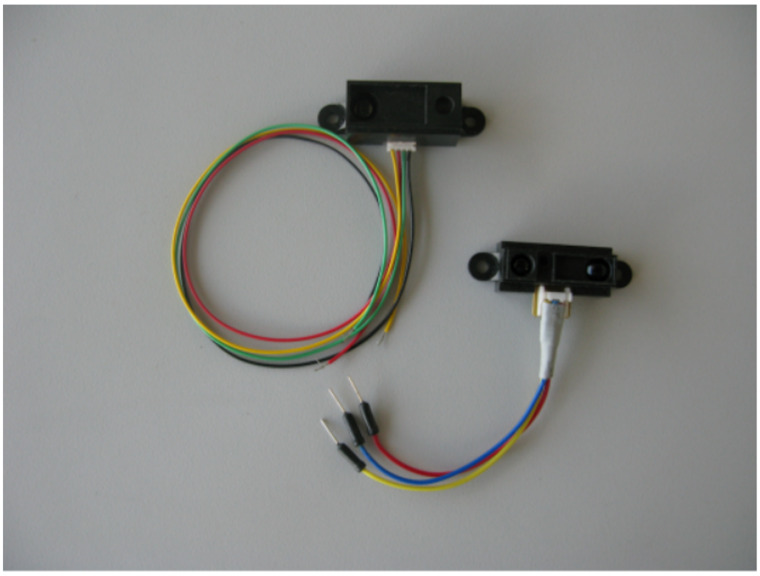

Figure 4: Sharp GP2D02 GP2D12 IR sensors

The Pioneer 3 robot that we will be simulating has a skirt of IR Range sensors.

IR Sensors

They allow us to find the distance between the object and the robot. An IR emitter sends an IR wave and we can calculate the distance to the obstacle by using the angle of the reflected IR wave.

Ultrasonic Sensors

They work by emitting ultrasound waves and uses the time taken for the echo to reach the receiver to calculate the distance between the object and the robot. They are typically more accurate than IR sensors and have a greater range.

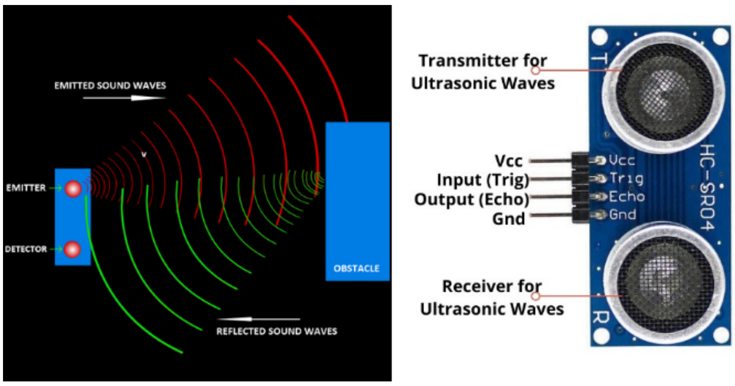

Figure 5: Ultrasonic sensor

LiDAR and RGB-D cameras are more sophisticated sensors. We will not be using them in our course. But you can read more about them with the links provided in the reference section.

Apart from detecting obstacles, our robot also has to reach a given target location. Assuming the robot knows the target location with respect to the start location, it is important to keep track of the robot's position when it moves around the environment. The process of using data from motion sensors to estimate change in position of robot over time is called odometry. Almost every vehicle has a odometer which keeps track of distance travelled. Odometry data is gathered using various sensors, one of the simplest of which we will be using are wheel encoders.

Encoders

Wheel encoders help us keep track of the angular movement of the wheel. When the radius of the wheel is known, we can easily calculate how much each wheel has moved using the angular movement of the wheels. Now some of you might ask, why do we need encoders? Won't we already know how much we move each wheel? The answer to this question is that no system is perfect. It is subject to inaccuracies due to various uncertainties. We might send control signals to a motor to make the wheel rotate by 30 degrees. There is a possibility that, due to a weak battery, the control signal (mostly current signals) could drop in value causing the motor to only rotate by 10 degrees. Therefore it is important to employ feedback sensors such as encoders. As to how encoders work, Refer to this document.

Useful Links

1. IR Sensors - Article
2. Differential drive robot - Article
3. Sensors - DroneBot Workshop - YouTube playlist